REMARKS

The Examiner contends that the provisional application upon which priority is claimed fails to provide adequate support under 35 U.S.C. § 112 for claims 1-6, 11-12, and 15 of this application for the reasons set forth in the previous Office Action and that claims to SEQ ID NOS: 1-7 of the instant application are, therefore, not supported by provisional application 60/048,810, and thus receive the priority date of the instant application, June 5, 1998.

The Examiner states that Applicant's response argues that nucleotide 5-419 of SEQ ID NO: 7 are the same as residues 1-415 in SEQ ID NO: 9 of the provisional application, but alleges that this is not persuasive because the claims are not limited to the portions of the sequences which are argued to be identical. Therefore, a new claim 34 has been added which specifically claims nucleotides 5-419 of SEQ ID NO: 7 and is entitled to the provisional priority date. Since the Examiner has not again raised the objection based on priority in her last Office Action regarding SEQ ID NO. 15 of the present application, Applicant infers that, as argued in Applicant's last response, SEQ ID NO. 15 received the provisional priority date of June 5, 1997.

Claim 14 stands rejected by the Examiner under 35 U.S.C. § 101 because the claimed invention is directed to non-statutory subject matter, for the reasons set forth in the previous Office Action, but she concludes that amending the claim to recite the purified polynucleotide, or similar language as supported by the specification as originally filed, may be sufficient to obviate this rejection. Therefore, Applicant has amended claim 14 as suggested by the Examiner.

Claim 14 also stands rejected by the Examiner under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out, and distinctly claim, the subject matter which Applicant regards as the invention for the reasons set forth in the previous Office Action.

Therefore, Applicant submits the software manual to the Wisconsin Sequence Analysis program, Version 8, publicly available from Genetics Computer Group, Madison, WI, as Exhibit A. Support for this submission is found on page 13, beginning on line 3. The manual provides the algorithm, parameters, parameter values and other information necessary to, accurately and consistently, calculate the percent identity. This manual indicates on pages 5-21, *inter alia,* that the software used the local homology

algorithm of Smith and Waterman (Advances in Applied Mathematics 2; 482-489 (1981)).

Claims 1-4, 22-25, 12, 30-33 are rejected under 35 U.S.C. § 112, first paragraph as containing subject matter which was not described in the specification. Specifically, the Examiner states the claims recite a "polynucleotide that specifically binds to a polynucleotide sequence", and that this language is not acceptable. Thus, in an effort to expedite prosecution, Applicant has deleted this language from the claims.

Claims 5-6 and 16-18 are rejected by the Examiner under 35 U.S.C. § 112, first paragraph, as containing subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art. The Examiner states that the newly amended claims now recite an open reading frame of "at least 5 amino acids", "at least 8 amino acids", and "at least 10 amino acids", and that the Specification discloses that an open reading frame (ORF) comprises the ranges of "at least about 3-5 amino acids", "at least about eight tot en amino acids" and "at least about 15-20 amino acids" and that the narrower limitation inserted into the claim is not supported by the broader teachings in the specification.

Applicant vigorously, yet respectfully, disagrees. The limitations in the claims are encompassed and taught in the Specification. The disclosure on page 17, lines 13-16 of the Specification include ORF's that are approximately 3, 4 or 5 amino acids long. "At least about 3-5 amino acids" clearly encompasses an ORF that is at least 5 amino acids long. Likewise, "at least about eight to ten amino acids" clearly discloses an ORF that is at least eight amino acids long. Based on the aforementioned, Applicant requests that the Examiner withdraw her rejection.

Claims 11, 22-25 and 30-33 are also rejected under 35 U.S.C. § 112, first paragraph, as containing subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art. The Examiner states that the newly amended claims now recite a polynucleotide of "at least about 10 nucleotides", "at least about 12 nucleotides", "at least about 15 nucleotides", or "at least about 20 nucleotides", alleging that the narrower limitation inserted into the claim is not supported by the broader teachings in the specification. Applicant refers the Examiner to the aforementioned argument relating to claims 5-6 and 16-18 in the preceding paragraph. For these same reasons, Applicant respectfully requests that the Examiner withdraw her rejection.

Claims 1-4 and 22-25 are rejected as being indefinite for reciting "polynucleotide sequence" as it is not clear how "sequences" could specifically bind to a polynucleotide (molecule). Therefore, in an effort to expedite prosecution, Applicant has deleted this language, thereby obviating this rejection.

Claims 1-4, 11, 12, and 22-25 are also rejected as indefinite for reciting "complement", but that by replacing the term "complement" with the phrase "full complement" or some other language that is supported by the specification as originally filed, this rejection would be overcome. Thus, Applicant has amended the claims as suggested by the Examiner, thereby overcoming this rejection.

Claims 1-4, 12, 22-25 and 30-33 are rejected as being indefinite" specifically binds to". The claims, as amended, delete this language, thereby obviating this rejection.

Claims 4, 11, 12, 19-21 and 27-29 are also rejected as being indefinite for reciting "epitope". The Examiner states that an epitope is defined as a portion of a molecule bound by a particular antibody and because the claims are silent as to the antibody which binds the epitope, it is impossible for one skilled in the art to determine the metes and bounds of the claims.

Applicant responds that methods for identifying epitopes in a novel peptide sequence are well known and described in both the scientific, commercial, and patent literature. For example, M. H. Van Regenmortel describes how to predict epitopes from the primary sequence of a protein. "Protein structure and antigenicity", *Int J Rad Appl Instrum B.*, **14(4)**:277-80 (1987).

Perkin-Elmer Biosystems, a major provider of DNA sequencing and peptide synthesizing instruments has established a public website which further describes how to select peptides which reflect the epitopes of a protein. (see http://www.pebio.com/pa/340913/html/chapt2.html#Choosing the Epitope.) This electronic publiction was posted in 1996 and basically describes the process employed by the Applicant of the current patent application.

Patent application PCT/US97/00485 also describes in detail how to identify epitopes from peptide sequences. The sequence can be scanned for hydrophobicity and hydrophilicity values by the method of Hopp, Prog. Clin. Biol. Res. 172B: 367-377 (1985) or the method of Cease et al, J. Exp. Med. 164: 1779-1784 (1986) or the method of Spouge et al, J. Immunol. 138: 204-212 (1987). Commercial software programs to implement these methods are available.

The Applicant further states that all of these well-known methods can be applied to all of the sequences of the current invention and every permutation and combination thereof. The success of the extant examples suggest that the results can be extrapolated to the entire sequence in identifying peptides which are useful for generating antibodies against the entire protein. Based on the aforementioned, it is respectfully requested that the Examiner withdraw her rejection.

Claims 5-6 and 16-18 are further rejected as being indefinite for reciting "a recombinant expression system", because it is not clear whether these claims are reciting a product, a kit or a method/process and for reciting an open reading frame operably linked to a control sequence. The Examiner suggests amending the claim to recite a promoter or enhancer or other such language as supported by the specification, as originally filed, may be sufficient to obviate this portion of the rejection. Applicant has amended the claims accordingly, thereby obviating this rejection.

Claims 11 and 19-21 are rejected by the Examiner as being indefinite for reciting "cell transformed with a nucleic acid sequence" because sequence is information and it is not clear how a cell can be transfected with information. Applicant has amended the claims to delete "nucleic acid" and substituted "polynucleotide", thereby overcoming this rejection.

The Examiner rejects claim 14 as indefinite for reciting a "gene" because it is not clear what structural features are encompassed by the claims. Thus, the Applicant has amended this claim to delete "gene" language and substitute "polynucleotide" as suggested by the Examiner on page 13.

The Examiner also rejects claims 1-6, 11, 12, 14, 16-18, 22-25 and 30-33 under 35 U.S.C. § 112, first paragraph, because the specification, while being enabling for polynucleotides comprising or consisting of SEQ ID NO: 1, 2, 3 or 7, does not reasonably provide enablement for the various fragments, genes, complements and polynucleotides which specifically bind to the various polynucleotides, as claims. Based on the present amendments to the claims, which delete "specifically binds" language, this rejection is deemed moot.

Claims 5, 6, and 16-18 are further rejected under 35 U.S.C. § 101 because the claims recite a recombinant expression system, however, it is not clear whether the claimed "system" is a product, a kit or a method. Applicant has, therefore, amended the

claims to delete the word "system" and include the term "vector", thus obviating this rejection.

The Examiner has also rejected claims 1-4, 12, 22-25 and 30-33 under 35 U.S.C. § 102(b) as being anticipated by Adams, *et al., Nature,* **377**:3-174 (1995), alleging Adams, *et al.,* teach a polynucleotide sequence that includes the claimed sequences that would be complements of SEQ ID NO: 1 or 3. The claims, as amended, cover full complements, not partial complements and Adams does not teach a full complement of SEQ ID NOS 1 or 3. Adams merely teaches a small portion of these sequences, and therefore, can not serve to anticipate the entire sequences in SEQ ID NOS. 1 or 3.

The Examiner states that claims 2 and 3 are drawn to a polynucleotide produced by recombinant or synthetic techniques and the method in which the polynucleotides were produced is immaterial to their patentability and, therefore, the Examiner also rejects claims 2 and 3 over Hillier, *et al.,* under 35 U.S.C. § 102(b). Since the composition of matter claims of the present application already cover nucleic acids, which are made from either recombinant or synthetic techniques, Applicant has canceled claims 2- and 3.

The Examiner states that claim 4 is drawn to polynucleotides that encode at least one epitope and for the purposes of this rejection, an epitope is considered as generally and well known to one of ordinary skill in the art as an amino acid sequence of a protein, usually 5-6 amino acids in length, that binds to an antibody. The Examiner concludes that considering the sequence identity of Adams, *et al.,* to SEQ ID NO: 1 or 3, taken in view of the breadth of the claims, there is at least one epitope encoded by the sequence taught by Adams, *et al.,* and that the fragments or complements recited in claims 12, and 22-25 encompass the polynucleotides of Adams, *et al.*

Based on the amendment to claim 1 of which 4 is dependent, and the aforementioned arguments relating to claim 1 (SEQ ID NOS 1 and 3), this rejection is deemed moot and the Examiner is respectfully requested to withdraw her rejection.

Claims 1-5, 11, 12, 16-18, 22-25 and 30-33 are rejected by the Examiner under 35 U.S.C. § 102(e) as being anticipated by Kuroda, *et al.,* (U.S. Patent 5,773,688 filed April 7, 1995), alleging Kuroda, *et al.,* teach polynucleotide molecules which encompass the fragments, derivatives, and complements recited in the claims.

Again, Kuroda, *et al.,* teach only a small segment of SEQ ID NO: 2, i.e., from position 15-64, not the entire sequence encompassed by the claim. Without the fragment language in the claims, this reference can not anticipate the claimed sequences.

It is appreciated that the Examiner has withdrawn: her rejection of claims 1-4 and 12 under 35 U.S.C. § 102(b) as being anticipated by Hillier, *et al.,* (GenBank Accession T94049); the rejection of claims 1-3 under 35 U.S.C. § 102(b) as being anticipated by NEB catalog; and the rejection of claims 5, 6 and 11 under 35 U.S.C. § 103(a) as being unpatentable over Hiller, *et al.* (GenBank Accession T94049) in view of Ausubel, *et al.*

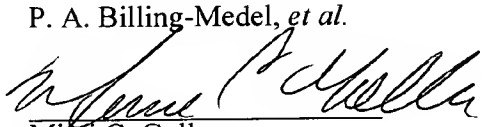
It is further appreciated that claim 26 is in condition for allowance, as indicated by the Examiner.

CONCLUSION

In view of the aforementioned amendments and remarks, the aforementioned application is in condition for allowance and Applicant requests that the Examiner withdraw all outstanding objections and rejections and to pass this application to allowance.

Respectfully submitted,

P. A. Billing-Medel, *et al.*



Mimi C. Goller
Registration No. 39,046
Attorney for Applicants

Abbott Laboratories
D377/AP6D-2
100 Abbott Park Road
Abbott Park, IL 60064-6050
(847) 935-7550
Fax: (847) 938-2623

EXHIBIT A

FUNCTION

BestFit makes an optimal alignment of the best segment of similarity between two sequences. Optimal alignments are found by inserting gaps to maximize the number of matches using the *local homology* algorithm of Smith and Waterman.

DESCRIPTION

BestFit inserts gaps to obtain the optimal alignment of the best region of similarity between two sequences, and then displays the alignment in a format similar to the output from Gap. The sequences can be of very different lengths and have only a small segment of similarity between them. You could take a short RNA sequence, for example, and run it against a whole mitochondrial genome.

SEARCHING FOR SIMILARITY

BestFit is the most powerful method in the Wisconsin Sequence Analysis Package™ for identifying the best region of similarity between two sequences whose relationship is unknown.

EXAMPLE

The sequence gamma.seq contains an Alu family sequence somewhere in the first 500 bases. alu.seq contains a generic human Alu family repeat. The two sequences are aligned and the best segment of similarity is found with BestFit.

```
% bestfit

 BESTFIT of what sequence 1 ?  gamma.seq

                  Begin (* 1 *) ?
                End (* 11375 *) ?  500
              Reverse (* No *) ?

 to what sequence 2 (* gamma.seq *) ?  alu.seq

                  Begin (* 1 *) ?
                End (*   207 *) ?
              Reverse (* No *) ?

 What is the gap creation penalty (* 5.00 *) ?

 What is the gap extension penalty (* 0.30 *) ?

 What should I call the paired output display file (* gamma.pair *)

 Aligning ..........-..

          Gaps:      3
       Quality:  129.3
 Quality Ratio:  0.625
  % Similarity:  84.466
        Length:    209

%
```

OUTPUT

Here is the output file. Notice how BestFit finds and displays only the best segments of similarity:

```
 BESTFIT of: gamma.seq  check: 6474  from: 1  to: 500

Human fetal beta globins G and A gamma
from Shen, Slightom and Smithies,  Cell 26; 191-203.
Analyzed by Smithies et al. Cell 26; 345-353.

 to: alu.seq  check: 4238  from: 1  to: 207

HSREP2 from the EMBL data library
Human Alu repetitive sequence located near the insulin gene
Dhruva D.R., Shenk T., Subramanian K.N.; "Integration in vivo into
Simian virus 40 DNA of a sequence that resembles a certain family of
genomic interspersed repeated sequences"; Proc. Natl. Acad. Sci. USA
77:4514-4518(1980).  . . .

 Symbol comparison table: Gencoredisk:[Gcgcore.Data.Rundata]Swgapdna.Cmp
 CompCheck: 5234

        Gap Weight:  5.000      Average Match:  1.000
     Length Weight:  0.300   Average Mismatch: -0.900

           Quality:  129.3             Length:    209
             Ratio:  0.625               Gaps:      3
Percent Similarity: 84.466   Percent Identity: 84.466

gamma.seq x alu.seq       June 20, 1994 15:15  ..

    137 AGACCAACCTGGCCAACATGGTGAAATCCCATCTCTAC.AAAAATACAAA 185
        |||||| ||||||||||||||||||| ||||||||||| ||||||||||
      1 AGACCAGCCTGGCCAACATGGTGAAACTCCATCTCTACTGAAAATACAAA 50

    186 AATTAGACAGGCATGATGGCAAGTGCCTGTAATCCCAGCTACTTGGGAGG 235
        |||||| |||||||| ||   ||||||| ||||||||||||| |||||
     51 AATTAGCCAGGCATGGTGATGCGTGCCTGGAATCCCAGCTACTTAGGAGG 100

    236 CTGAGGAAGGAGAATTGCTTGAACCTGGAAGGCAGGAGTTGCAGTGAGCC 285
        ||||| || ||||| ||| |||| | ||| | ||||||||||||
    101 CTGAGACAGAAGAATCCCTTAAACCAAG.AGGTGGAGGTTGCAGTGAGCC 149

    286 GAGATCATACCACTGCACTCCAGCCTGGGTGACAGAACAAGACTCTGTCT 335
        |||||| || |||||||||||||| ||||||||| | ||||||  |||
    150 GAGATCGCACGGCTGCACTCCAGCCT.GGTGACAGAGCGAGACTCCATCT 198

    336 CAAAAAAAA 344
        |||||||||
    199 CAAAAAAAA 207
```

RELATED PROGRAMS

When you want an alignment that covers the whole length of both sequences, use Gap. When you are trying to find only the best segment of similarity between two sequences, use BestFit. PileUp creates a multiple sequence alignment of a group of related sequences, aligning the whole length of all sequences. DotPlot displays the entire surface of comparison for a comparison of two sequences. GapShow displays the pattern of differences between two aligned sequences. PlotSimilarity plots the average similarity of two or more aligned sequences at each position in the alignment. Pretty displays alignments of several sequences. LineUp is an editor for editing multiple sequence alignments. CompTable helps generate scoring matrices for peptide comparison.

ALGORITHM

BestFit uses the *local homology* algorithm of Smith and Waterman (Advances in Applied Mathematics 2; 482-489 (1981)) to find the best segment of similarity between two sequences. BestFit reads a scoring matrix that contains values for every possible GCG symbol match (see the LOCAL DATA FILES topic below). The program uses these values to construct a path matrix that represents the entire surface of comparison with a score at every position for the best possible alignment to that point. The *quality* score for the best alignment to any point is equal to the sum of the scoring matrix values of the matches in that alignment, less the gap creation penalty times the number of gaps in that alignment, less the gap extension penalty times the total length of all gaps in that alignment. The gap creation and gap extension penalties are set by you. If the best path to any point has a negative value, a zero is put in that position.

After the path matrix is complete, the highest value on the surface of comparison represents the end of the best region of similarity between the sequences. The best path from this highest value backwards to the point where the values revert to zero is the alignment shown by BestFit. This alignment is the best segment of similarity between the two sequences.

For nucleic acids, the default scoring matrix has a *match* value of 1.0 for each identical symbol comparison and -0.90 for each non-identical comparison (not considering nucleotide ambiguity symbols for this example). The *quality* score for a nucleic acid alignment can, therefore, be determined using the following equation:

```
Quality = 1.0 x TotalMatches +  -0.90 x TotalMismatches
                  - (GapCreationPenalty x GapNumber)
          - (GapExtensionPenalty x TotalLengthOfGaps)
```

The *quality* score for a protein alignment is calculated in a similar manner. However, while the default nucleic acid scoring matrix has a single value for all non-identical comparisons, the default protein scoring matrix has different values for the various non-identical amino acid comparisons. The *quality* score for a protein alignment can therefore be determined using the following equation (where $Total_{AA}$ is the total number of A-A (Ala-Ala) matches in the alignment, $CmpVal_{AA}$ is the value for an A-A comparison in the scoring matrix, $Total_{AB}$ is the total number of A-B (Ala-Asx) matches in the alignment, $CmpVal_{AB}$ is the value for an A-B comparison in the scoring matrix, ...) :

$$
\begin{aligned}
Quality = {} & CmpVal_{AA} \times Total_{AA} \\
& + CmpVal_{AB} \times Total_{AB} \\
& - CmpVal_{AC} \times Total_{AC} \\
& \\
& - CmpVal_{ZZ} \times Total_{ZZ} \\
& - (GapCreationPenalty \times GapNumber) \\
& - (GapExtensionPenalty \times TotalLengthOfGaps)
\end{aligned}
$$

For a more complete discussion of scoring matrices, see the **Data Files** manual.

CONSIDERATIONS

BestFit Always Finds Something

BestFit always finds an alignment for any two sequences you compare -- even if there is no significant similarity between them! You must evaluate the results critically to decide if the segment shown is not just a random region of relative similarity.

The Segments Shown Obscure Alternative Segments

BestFit only shows one segment of similarity, so if there are several, all but one is obscured. You can approach this problem with graphic matrix analysis (see the Compare and DotPlot programs). Alternatively, you can run BestFit on ranges outside the ranges of similarity found in earlier runs to bring other segments out of the shadow of the best segment.

The Best Fit is Only One Member of a Family

Like all fast gapping algorithms, the alignment displayed is a member of the family of best alignments. This family may have other members of equal quality, but will not have any member with a higher quality. The family is usually significantly different for different choices of gap creation and gap extension penalties. See the CONSIDERATIONS topic in the entry for the Gap program in the **Program Manual** to learn more about how to assign gap creation and gap extension penalties.

The Surface of Comparison

The magnitude of the computer's job is proportional to the area of the surface of comparison. That area is determined by the product of the lengths of the two sequences compared. BestFit can evaluate a surface of up to 3.5 million elements. This surface would be large enough to compare two sequences approximately 1,870-symbols long, or one sequence 200-symbols long with another sequence 17,500-symbols long. When you have much longer sequences that are known to align well, you can use the command-line option `-LIMit` to use the surface more efficiently.

The Public Scoring Matrix for Nucleic Acid Comparisons is Very Stringent

The scoring matrix swgapdna.cmp penalizes mismatches -0.9 so the segments found may be very brief. This penalty means that the alignment cannot be extended by three bases to pick one extra match. The scoring matrix used by Smith and Waterman, when local alignments were first described, used -0.333 for the mismatch penalty. You can use Fetch to copy randomdna.cmp and rename it swgapdna.cmp to use these values, or use nwsgapdna.cmp, which has no mismatch penalty at all.

Rapid Alignment

When possible, BestFit tries to find the optimal alignment very quickly. If this rapid alignment is not unambiguously optimal, BestFit automatically realigns the sequences to calculate the optimal alignment. When this occurs, the monitor of alignment progress on your terminal screen (`Aligning...`) is displayed twice for a single alignment.

ALIGNING LONG SEQUENCES

This program can align very long sequences if you know roughly where the alignment of interest begins. Run the program with the command line option `-LIMit`. Then set the starting coordinates for each sequence near the point where the alignment of interest begins and set gap shift limits on each sequence. The program then aligns the sequences from your starting point such that the sequences do not get out of phase by more than the gap shift limits you have set. If you started both sequences at

base number one and set the gap shift limit for sequence one to 100 and for sequence two to 50, then base 350 in sequence one could not be gapped to any base outside of the range from 300 to 450 on sequence two.

If you omit `-LIMit` on the command line, the program automatically sets gap shift limits if they are needed to allow the alignment of long sequences to proceed. In this case, the program limits the total length of gaps that can be inserted into each sequence and calculates the best alignment within this incomplete, or *limited*, surface of comparison. The program then performs a calculation to determine whether the alignment could possibly be improved if there were no restriction on the total length of gaps in each sequence. If the program cannot rule out this possibility, it displays the message `*** Alignment is not guaranteed to be optimal ***`. Because the criteria used in the calculation for guaranteeing an optimal alignment are very stringent, a limited alignment often may be optimal even if this message is displayed. In any event, the program continues to completion.

EVALUATING ALIGNMENT SIGNIFICANCE

This program can help you evaluate the significance of the alignment, using a simple statistical method, with the `-RANdomizations` command line option. The second sequence is repeatedly shuffled, maintaining its length and composition, and then realigned to the first sequence. The average alignment score, plus or minus the standard deviation, of all randomized alignments is reported in the output file. You can compare this average *quality* score to the quality score of the actual alignment to help evaluate the significance of the alignment. The number of randomizations can be specified along with the `-RANdomizations` command line qualifier; the default is 10.

The score of each randomized alignment is reported to the screen. You can use `<Ctrl>C` to interrupt the randomizations and output the results from those randomized alignments that have been completed.

By ignoring the statistical properties of biological sequences, this simple Monte Carlo statistical method may give misleading results. Please see Lipman, D.J, Wilbur, W.J., Smith, T.F., and Waterman, M.S. (Nucl. Acids Res. 12; 215-226 (1984)) for a discussion of the statistical significance of nucleic acid similarities.

ALIGNMENT METRICS

BestFit and Gap display four figures of merit for alignments: Quality, Ratio, Identity, and Similarity.

The Quality (described above) is the metric maximized in order to align the sequences. Ratio is the quality divided by the number of bases in the shorter segment. Percent Identity is the percent of the symbols that actually match. Percent Similarity is the percent of the symbols that are similar. Symbols that are across from gaps are ignored. A similarity is scored when the scoring matrix value for a pair of symbols is greater than or equal to 0.50, the *similarity threshold*. This threshold is also used by the display procedure to decide when to put a ':' (colon) between two aligned symbols. You can reset it from the command line with the second optional parameter of `-PAIr`. For instance, the expression `-PAIr=1.0,0.5` would set the similarity threshold to 0.5.

The similarity and identity metrics are not optimized by alignment programs so they should not be used to compare alignments.

PEPTIDE SEQUENCES

If your input sequences are peptide sequences, this program uses a scoring matrix with matches scored as 1.5 and mismatches scored according to the evolutionary distance between the amino acids as measured by Dayhoff and normalized by Gribskov (Gribskov and Burgess Nucl. Acids Res. 14(16); 6745-6763 (1986)).

RESTRICTIONS

Input sequences may not be more than 30,000-symbols long. This program cannot evaluate a surface of comparison larger than 5.5 million elements. A 200 x 27,500 comparison is possible, as well as a 2,300 x 2,300 comparison. See the ALIGNING LONG SEQUENCES topic for help in aligning long sequences that would normally exceed the maximum surface of comparison. You can also ask your system manager to increase the maximum surface of comparison if your system has enough virtual memory.

SEQUENCE TYPE

The function of BestFit depends on whether your input sequence(s) are protein or nucleotide. Normally the type of a sequence is determined by the presence of either `Type: N` or `Type: P` on the last line of the text heading just above the sequence itself. If your sequence(s) are not the correct type, turn to Appendix VI for information on how to change or set the type of a sequence.

COMMAND-LINE SUMMARY

All parameters for this program may be put on the command line. Use the option `-CHEck` to see the summary below and to have a chance to add things to the command line before the program executes. In the summary below, the capitalized letters in the qualifier names are the letters that you *must* type in order to use the parameter. Square brackets ([and]) enclose qualifiers or parameter values that are optional. For more information, see "Using Program Parameters" in Chapter 3, Basic Concepts: Using Programs in the **User's Guide**.

```
Minimal Syntax: % bestfit [-INfile1=]gamma.seq  [-INfile2=]alu.seq -Default

Prompted Parameters:

-BEGin1=1 -BEGin2=1       beginning of each sequence
-END1=500 -END2=207       end of each sequence
-NOREV1     -NOREV2       strand of each sequence
-GAPweight=5.0            gap creation penalty     (3.0 is protein default)
-LENgthweight=0.3         gap extension penalty    (0.1 is protein default)
[-OUTfile1=]gamma.pair    output file for alignment

Local Data Files: -DATa=swgapdna.cmp  scoring matrix for nucleic acids
                  -DATa=swgappep.cmp  scoring matrix for peptides

Optional Parameters:

-OUTfile2=gamma.gap       new sequence file for sequence 1 with gaps added
-OUTfile3=alu.gap          "     "       "   "      "    2   "   "     "
-LIMit1=499 -LIMit2=206 limit the surface of comparison
-RANdomizations[=10]      determine average score from 10 randomized
                             alignments
-PAIr=1.0,0.5,0.1         thresholds for displaying '|', ':', and '.'
-WIDth=50                 the number of sequence symbols per line
-PAGe=60                  adds a line with a form feed every 60 lines
-NOBIGGaps                suppresses abbreviation of large gaps with '.'s
-HIGhroad                 makes the top alignment for your parameters
-LOWroad                  makes the bottom alignment for your parameters
-NOSUMmary                suppresses the screen summary
```

ACKNOWLEDGEMENTS

Gap and BestFit were originally written for Version 1.0 by Paul Haeberli from a careful reading of the Needleman and Wunsch (J. Mol. Biol. **48**; 443-453 (1970)) and the Smith and Waterman (Adv. Appl. Math. **2**; 482-489 (1981)) papers.

Limited alignments were designed by Paul Haeberli and added to the Package for Version 3.0. They were united into a single program by Philip Delaquess for Version 4.0. Default gap penalties for protein alignments were modified according to the suggestions of Rechid, Vingron and Argos (CABIOS **5**; 107-113 (1989)).

LOCAL DATA FILES

The files described below supply auxiliary data to this program. The program automatically reads them from a public data directory unless you either 1) have a data file with exactly the same name in your current working directory; or 2) name a file on the command line with an expression like `-DATa1=myfile.dat`. For more information see Chapter 4, Using Data Files in the **User's Guide.**

If the first sequence you name is a nucleic acid, BestFit uses the scoring matrix in the public file swgapdna.cmp. (SW stands for Smith and Waterman.) If the first sequence you name is a peptide sequence, BestFit reads swgappep.cmp instead. The presence of these files in your current working directory causes BestFit to read your version instead. (See the **Data Files** manual for more information about scoring matrices.)

OPTIONAL PARAMETERS

The parameters and switches listed below can be set from the command line. For more information, see "Using Program Parameters" in Chapter 3, Basic Concepts: Using Programs in the **User's Guide.**

`-LIMit1=20` and `-LIMit2=20`

let you set *gap shift limits* for each sequence. When you already know of a long similarity between two sequences you can "zip" them together using this mode. The beginning coordinates for each sequence must be near the beginning of the alignment you want to see. The alignment continues so that gaps inserted do not require the sequences to get out of step by more than the gap shift limits. You can align very long sequences rapidly. The surface of comparison is still limited to 3.5 million. The size of a comparison can be predicted by multiplying the average length of the two sequences by the sum of the two shift limits.

If you add `-LIMit` to the command line without any qualifier value, the program prompts you to enter gap shift limits for each sequence.

`-RANdomizations=10`

reports the average alignment score and standard deviation from 10 randomized alignments in which the second sequence is repeatedly shuffled, maintaining the length and composition of the original sequence, and then aligned to the first sequence. You can use the optional parameter to set the number of randomized alignment to some number other than 10.

`-OUTfile2=seqname1.gap -OUTfile3=seqname2.gap`

This program can write three different output files. The first displays the alignment of sequence one with sequence two. The second is a new sequence file for sequence one, possibly expanded by gaps to make it align with sequence two. The third, like the second, is a new sequence file for sequence two, possibly expanded by gaps to make it align with sequence one. The program writes only the first file unless there are output file options on the command line. If there are any output files named on the command line, *only* those output files are written. If you add

-OUT to the command line without any qualifying filename, then the program will write the second and third output files after prompting you for their names.

Aligned sequences (in sequence files) can be displayed with GapShow. Their similarity can be displayed with PlotSimilarity.

-PAIr=1.0,0.5,0.1

The paired output file from this program displays sequence similarity by printing one of three characters between similar sequence symbols: a pipe character(|), a colon (:), or a period (.). Normally a pipe character is put between symbols that are the same, a colon is put between symbols whose comparison value is greater than or equal to 0.50, and a period is put between symbols whose comparison value is greater than or equal to 0.10. You can change these *match display thresholds* from the command line. The three parameters for **-PAIr** are the display thresholds for the pipe character, colon, and period. The match display criterion for a pipe character changes from symbolic identity (the default) to the quantitative threshold you have set in the first parameter. A pipe character will no longer be inserted between identical symbols unless their comparison values are greater than or equal to this threshold. If you still want a pipe character to connect identical symbols, use **x** instead of a number as the first parameter. (See the **Data Files** manual for more information about scoring matrices.)

-PAGe=64

When you print the output from this program, it may cross from one page to another in a frustrating way -- especially when you print on individual sheets. This option adds form feeds to the output file in order to try to keep clusters of related information together. You can set the number of lines per page by supplying a number after the **-PAGe** qualifier.

-WIDth=50

puts 50 sequence symbols on each line of the output file. You can set the width to anything from 10 to 150 symbols.

-NOBIGGaps

suppresses large gap abbreviations, showing all the sequence characters across from large gaps. Usually, gaps that extend one sequence by more than one complete line of output are abbreviated with three dots arranged in a vertical line.

-LOWroad and -HIGhroad

The insertion of gaps is, in many cases, arbitrary, and equally optimal alignments can be generated by inserting gaps differently. When equally optimal alignments are possible, this program can insert the gaps differently if you select either the **-LOWroad** or the **-HIGhroad** options. Here are examples for the alignment of GACCAT with GACAT with different parameters.

```
For:        Match = 1.0        MisMatch = -0.9
      Gap weight = 1.0   Length Weight =  0.0

LowRoad:   1 GACCAT 6
             :  |||      Quality = 4.0
           1 GA.CAT 5

HighRoad:  1 GACCAT 6
             |||  ||     Quality = 4.0
           1 GAC.AT 5
```

```
For:        Match = 1.0        MisMatch =  0.0
      Gap weight = 3.0   Length Weight =  0.0

HighRoad:  1 GACCAT 6
             |||           Quality = 3.0
           1 GACAT. 5

LowRoad:   1 GACCAT 6
             ...|||        Quality = 3.0
           1 .GACAT 5
```

Essentially the *low road* shifts all of the arbitrary gaps in sequence two to the left and all of the arbitrary gaps in sequence one to the right. The *high road* does exactly the opposite. When neither *high road* nor *low road* is selected, the program tries not to insert a gap whenever that is possible and uses the high road alternative for all collisions.

-SUMmary

writes a summary of the program's work to the screen when you've used the `-Default` qualifier to suppress all program interaction. A summary typically displays at the end of a program run interactively. You can suppress the summary for a program run interactively with `-NOSUMmary`.

Use this qualifier also to include a summary of the program's work in the log file for a program run in batch.